



PY32F003 系列

32 位 ARM® Cortex®-M0+ 微控制器

HAL 库样例手册

---

**PY32F003 系列**

**32 位 ARM® Cortex®-M0+ 微控制器**

**HAL 库样例手册**

## 1 ADC

### 1.1 ADC\_AnalogWatchdog

此样例演示了 ADC 的模拟看门狗功能，当开启看门狗通道的电压值不在设定的上下限中，会进入看门狗中断。

### 1.2 ADC\_ContinuousConversion\_DMA

此样例演示了通过 ADC 触发 DMA 搬运 ADC\_DR 中的数据,配置 CH0(PA00)为 ADC 的模拟输入通道, ADC 触发 DMA 搬运采样数据, DMA 搬运完成, 在循环中打印搬运的数据。

### 1.3 ADC\_MultiChannelSingleConversion\_TriggerSW

此样例演示了 ADC 单次轮询(Polling)的方式采样 AN0,AN1,AN4 通道,并通过串口打印出来。

### 1.4 ADC\_SingleConversion\_TriggerTimer\_DMA

此样例演示了通过 TIM1 触发 ADC,再通过 ADC 触发 DMA 搬运 ADC\_DR 中的数据, 配置 CH0(PA00)为 ADC 的模拟输入通道, TIM1 配置为主模式, TIM1 每产生一次更新事件, 触发一次 ADC 采样, 然后 ADC 触发 DMA 搬运采样数据, 采样数据在中断中打印。

### 1.5 ADC\_SingleConversion\_TriggerTimer\_IT

此样例演示了通过 TIM1 触发 ADC 模块的通道采样功能, 配置 CH0(PA00)为 ADC 的模拟输入通道, TIM1 配置为主模式, TIM1 每产生一次更新事件, 触发一次 ADC 采样, 采样数据在中断中打印。

### 1.6 ADC\_TempSensor

此样例演示了 ADC 模块的温度传感器功能, 串口每隔 500ms 打印当前的温度。

### 1.7 ADC\_Vrefint

此样例演示了 ADC 模块的 VCC 采样功能, 通过采样 VREFINT 的值, 计算得出 VCC 的值, 并通过串口打印出来。

## 2 COMP

### 2.1 COMP\_CompareGpioVsVrefint\_HYST

此样例演示了 COMP 比较器迟滞功能，PA03 作为比较器正端输入，VREFINT 作为比较器负端输入，PA07 作为比较器的输出端口，通过调整 PA03 上的输入电压，观测 PA07 引脚上的电平变化。

### 2.2 COMP\_CompareGpioVsVrefint\_IT

此样例演示了 COMP 比较器功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，通过调整 PA01 上的输入电压，观测 PA06 引脚上的电平变化和 PB5 上的电压翻转。

### 2.3 COMP\_CompareGpioVsVrefint\_Polling

此样例演示了 COMP 比较器功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，通过调整 PA01 上的输入电压，观测 PA06 引脚上的电平变化。

### 2.4 COMP\_CompareGpioVsVrefint\_WakeUpFromStop

此样例演示了 COMP 比较器唤醒功能，PA01 作为比较器正端输入，VREFINT 作为比较器负端输入，PA06 作为比较器的输出端口，进入 stop 模式后，通过调整 PA01 上的输入电压，产生中断唤醒 stop 模式。

### 2.5 COMP\_CompareGpioVsVrefint\_Window

此样例演示了比较器窗口功能，PA01 作为比较器正端输入，内部通过 COMP1 和 COMP2 的正极相连，PA1 输入大于 VREF 电压，LED 以 200ms 的间隔进行翻转；PA1 输入小于  $1/4V_{REF}$  的电压，LED 熄灭；PA1 输入的电压小于 VREF 大于  $1/4V_{REF}$ ，LED 常亮。

## 3 CRC

### 3.1 CRC\_CalculateCheckValue

此样例演示了 CRC 校验功能，通过对一个数组里的数据进行校验，得到的校验值与理论校验值进行比较，相等则 LED 灯亮，否则 LED 灯熄灭。

## 4 DMA

### 4.1 DMA\_SramToSram

此样例演示了 DMA 从 SRAM 到 SRAM 传输数据的功能（SRAM 和外设之间传输的样例请参考相关外设样例工程）。

## 5 EXTI

### 5.1 EXTI\_ToggleLed\_IT

此样例演示了 GPIO 外部中断功能，PA12 引脚上的每一个下降沿都会产生中断，中断函数中 LED 灯会翻转一次。

### 5.2 EXTI\_WakeUp\_Event

此样例演示了通过 PA6 引脚唤醒 MCU 的功能。下载程序并运行后，LED 灯处于常亮状态；按下用户按键后，LED 灯处于常暗状态，且 MCU 进入 STOP 模式；拉低 PA6 引脚后，MCU 唤醒，LED 灯处于闪烁状态。

## **6 FLASH**

### **6.1 FLASH\_OptionByteWrite\_RST**

此样例演示了通过软件方式将 RESET 引脚改为普通 GPIO。

### **6.2 FLASH\_PageEraseAndWrite**

此样例演示了 flash page 擦除和 page 写功能。

### **6.3 FLASH\_SectorEraseAndWrite**

此样例演示了 flash sector 擦除和 page 写功能。

## 7 GPIO

### 7.1 GPIO\_FastIO

本样例主要展示 GPIO 的 FAST IO 输出功能，FAST IO 速度可以达到单周期翻转速度。

### 7.2 GPIO\_Toggle

此样例演示了 GPIO 输出模式，配置 LED 引脚为数字输出模式，并且每隔 250ms 翻转一次 LED 引脚电平，运行程序，可以看到 LED 灯以 2Hz 的频率闪烁。



## 8 I2C

### 8.1 I2C\_TwoBoard\_CommunicationMaster\_DMA

此样例演示了 I2C 通过 DMA 方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据;主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

### 8.2 I2C\_TwoBoard\_CommunicationMaster\_DMA\_MEM

此样例演示了主机 I2C 通过 DMA 方式进行通讯，从机使用 EEPROM 外设芯片 P24C32，按下 user 按键，主机先向从机写 15bytes 数据为 0x1~0xf，然后再从 EEPROM 中将写入的数据读出，读取成功后，主机板上的小灯处于“常亮”状态。

### 8.3 I2C\_TwoBoard\_CommunicationMaster\_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据;主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

### 8.4 I2C\_TwoBoard\_CommunicationMaster\_Polling

此样例演示了 I2C 通过轮询方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据;主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

### 8.5 I2C\_TwoBoard\_CommunicationSlave\_DMA

此样例演示了 I2C 通过 DMA 方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据;主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

### 8.6 I2C\_TwoBoard\_CommunicationSlave\_IT

此样例演示了 I2C 通过中断方式进行通讯，主机先向从机发送 15byte 数据，然后再接收从机发送的 15byte 数据;主机、从机接收数据成功后，主机和从机板上的小灯处于“常亮”状态。

## 9 IWDG

### 9.1 IWDG\_Reset

此样例演示了 IWDG 看门狗功能，配置看门狗重载计数值，计数 1s 后复位，然后通过调整每次喂狗的时间（main 函数 while 循环中代码），可以观察到，如果每次喂狗时间小于 1s，程序能一直正常运行（LED 灯闪烁），如果喂狗时间超过 1s，程序会一直复位（LED 灯熄灭）。

## 10 LPTIM

### 10.1 LPTIM\_Wakeup

此样例演示了 LPTIM 中断唤醒 stop 模式，每次唤醒后再次进入 stop 模式，每 200ms 唤醒一次。

## **11 PWR**

### **11.1 PVD**

此样例演示了 PVD 电压检测功能，样例中配置 PB07 引脚的电压与 VREF(1.2v)进行比较，当 PB07 引脚的电压高于 VREF 时,LED 灯灭，当低于 VREF 时，LED 灯亮。

### **11.2 PWR\_SLEEP\_WFE**

此样例演示了 sleep 模式下，通过 GPIO 事件唤醒功能。

### **11.3 PWR\_SLEEP\_WFI**

此样例演示了 sleep 模式下，GPIO 外部中断唤醒功能。

### **11.4 PWR\_STOP\_WFE**

此样例演示了 stop 模式下，通过 GPIO 事件唤醒功能。

### **11.5 PWR\_STOP\_WFI**

此样例演示了 stop 模式下，GPIO 外部中断唤醒功能。

## 12 RCC

### 12.1 RCC\_HSEDiv

此样例配置系统时钟为 HSE，并通过 MCO (PA01) 引脚输出

### 12.2 RCC\_HSCalibration

此样例校准内部 HSI

### 12.3 RCC\_HSIOutput

此样例配置系统时钟为 HSI，并通过 MCO (PA01) 引脚输出

### 12.4 RCC\_LSIOutput

此样例配置系统时钟为 LSI，并通过 MCO (PA01) 引脚输出，注意系统时钟切换为 LSI 之前，要求把 systick 中断关闭掉，因为 systick 中断默认是 1ms 一次中断，由于 LSI 时钟频率过低，systick 中断会导致程序无法正常运行。

### 12.5 RCC\_SysclockSwitch

此样例演示系统时钟切换功能，样例中配置系统时钟从 LSI 切换到 HSE，并通过 MCO (PA01) 引脚输出系统时钟。

## 13 RTC

### 13.1 RTC\_AlarmSecond\_IT

此样例演示 RTC 的秒中断和闹钟中断功能，每次秒中断，在中断函数中会打印字符“RTC\_IT\_SEC”，并且输出实时时间。

### 13.2 RTC\_WakeUpSecond

此样例演示了通过 RTC 的秒中断唤醒 MCU 的功能。下载程序并运行后，LED 灯处于常亮状态；按下用户按键后，LED 灯处于常暗状态，且 MCU 进入 STOP 模式；RTC 秒中断唤醒 MCU 后，LED 灯处于闪烁状态。

## 14 SPI

### 14.1 SPI\_TwoBoards\_FullDuplexMaster\_DMA

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信 的演示,此接口设置为主模式,为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据,数据以主机提供的 SCK 沿同步被移位,完成全双工通信。

### 14.2 SPI\_TwoBoards\_FullDuplexSlave\_DMA

此样例是对串口外设接口（SPI）与外部设备以全双工串行方式进行通信 的演示,此接口设置为主模式,为外部从设备提供通信时钟 SCK。主机通过 MOSI 引脚发送数据,从 MISO 引脚接收从机的数据,数据以主机提供的 SCK 沿同步被移位,完成全双工通信。

## 15 TIMER1

### 15.1 TIM1\_AutoReloadPreload

此样例实现了定时器的基本计数功能，以及演示了 ARR 自动重载功能，样例在定时器重载中断中翻转 LED 灯 修改 main.c 中的第 56 行配置 TimHandle.Init.AutoReloadPreload = TIM\_AUTORELOAD\_PRELOAD\_ENABLE;使能自动重载功能，新的 ARR 值在第四次进中断时生效，配置 TimHandle.Init.AutoReloadPreload = TIM\_AUTORELOAD\_PRELOAD\_DISABLE;禁止自动重载功能，新的 ARR 值在第三次进中断时生效,生效后，LED 灯每隔 400ms 翻转一次

### 15.2 TIM1\_ComplementarySignals

此样例实现了定时器的互补输出功能，一组互补共两路 pwm 输出，此样例没有实现死区功能 CH1 -> PA3CH1N -> PA0

### 15.3 TIM1\_ComplementarySignals\_break

此样例实现了定时器的刹车功能，CH1 和 CH1N 互补 pwm 输出，接收到外部 IO 口的刹车信号（低电平）后，PWM 信号关闭，由于 BDTR.AOE 置位，所以刹车信号取消（高电平）后，继续 pwm 输出，此样例实现了死区功能。CH1 -> PA3CH1N -> PA0 刹车输入 -> PA6 通过调整 OCxE,CCxP,OISx,CCxNE,CCxNP,OISxN 的配置，可实现刹车功能的各种应用

### 15.4 TIM1\_ComplementarySignals\_break\_it

此样例实现了定时器的刹车功能，CH1 和 CH1N 互补 pwm 输出，接收到外部 IO 口的刹车信号（低电平）后，PWM 信号关闭，由于 BDTR.AOE 置位，所以刹车信号取消（高电平）后，继续 pwm 输出，此样例实现了死区功能。本样例开启了刹车中断，并在刹车中断里翻转 LED 灯 CH1 -> PA3CH1N -> PA0 刹车输入 -> PA6 通过调整 OCxE,CCxP,OISx,CCxNE,CCxNP,OISxN 的配置，可实现刹车功能的各种应用

### 15.5 TIM1\_ComplementarySignals\_DeadTime

此样例实现了定时器的刹车功能，CH1 和 CH1N 互补 pwm 输出，接收到外部 IO 口的刹车信号（低电平）后，PWM 信号关闭，由于 BDTR.AOE 置位，所以刹车信号取消（高电平）后，继续 pwm 输出，此样例实现了死区功能。CH1 -> PA3CH1N -> PA0 刹车输入 -> PA6 通过调整 OCxE,CCxP,OISx,CCxNE,CCxNP,OISxN 的配置，可实现刹车功能的各种应用



## 15.6 TIM1\_DmaBurst\_twice

此样例演示了在 TIM1 中使用 DMA 连续两次 burst 传输数据的功能, burst 每传输一次更新三个寄存器, PSC,ARR,RCR, 在更新事件中断中, PA0 会进行翻转, 通过逻辑分析仪监测, 可看到 PA0 的翻转间隔会从第一次的 400ms, 第二次 400ms, 第三次 10ms,第四次及后续变为 100us, 此时两次 burst 传输完成, 并且 PCS,ARR,RCR 均更新完毕。

## 15.7 TIM1\_ExternalClockMode1

此样例演示了 TIM1 的外部时钟模式 1 功能, 选择 ETR(PA12)引脚作为外部时钟输入源, 并使能更新中断, 在中断中翻转 LED 灯

## 15.8 TIM1\_ExternalClockMode2

此样例演示了 TIM1 的外部时钟模式 2 功能, 选择 ETR(PA12)引脚作为外部时钟输入源, 并使能更新中断, 在中断中翻转 LED 灯

## 15.9 TIM1\_InputCapture\_TI1FP1

此样例演示了在 TIM1(PA3)输入捕获功能, PA8 输入时钟信号, TIM1 捕获成功后, 会进入捕获中断, 每进一次中断, 翻转一次 LED

## 15.10 TIM1\_OCToggle

此样例演示了 TIM1 比较模式下的 OC 翻转输出功能, 使能 CH1(PA3),CH2(PA13),CH3(PA0),CH4(PA1) 四个通道的输出功能, 并且当计数器 TIMx\_CNT 与 TIMx\_CCRx 匹配时输出信号翻转, 周期为 200KHz

## 15.11 TIM1\_OnePulseOutput

此样例演示了 TIM1 的单脉冲模式, CH2(PA13)引脚上的上升沿, 触发计数器开始计数, 当计数值与 CCR1 匹配时, CH1(PA3)输出高电平, 直到计数器溢出, CH1 再次输出低电平, 计数器溢出后, 定时器停止工作。本例程脉冲宽度计算  $(TIM1\_ARR-TI1\_CCR1)/CLK = (65535-16383)/16000000 = 3.072ms$

## 15.12 TIM1\_SynchronizationEnable

定时器 1 的使能由定时器 3 控制, 当定时器 3 计数时, LED 会常亮, 当定时器 3 发生更新事件时, 更新事件会触发定时器 1, 定时器 1 开始计数后, LED 会以 5Hz 的频率进行翻转

### 15.13 TIM1\_TIM3\_Cascade

此样例实现了 TIM1 和 TIM3 级联成 32 位计数器，TIM3 做主机，TIM3 的计数溢出信号作为 TIM1 的输入时钟，通过配置 TIM1 和 TIM3 的重载寄存器值，(在 TIM1 中断回调函数中) 实现 LED 灯以 0.5Hz 频率闪烁。此例程计算方式为  $TIM3\_ARR * TIM3\_PSC * TIM1\_ARR * TIM1\_PSC / \text{时钟} = 800 * 100 * 100 * 1 / 8000000 = 1\text{Hz}$

### 15.14 TIM1\_Update\_DMA

此样例演示了在 TIM1 中使用 DMA 传输数据的功能，通过 DMA 从 SRAM 中搬运数据到 ARR 寄存器，实现 TIM1 周期变化，在 TIM1 第一次溢出后，PA0 会翻转，此时翻转间隔为 400ms，DMA 开始搬运数据到 TIM1\_ARR，第二次 PA0 翻转间隔为 400ms，第三次翻转间隔为 100ms，第四次翻转间隔为 200ms，第四次翻转间隔为 300ms，此时 DMA 搬运结束，后续翻转间隔均为 300ms。

### 15.15 TIM1\_Update\_IT

此样例演示了在 TIM1 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

## 16 TIMER16

### 16.1 TIM16\_Counter

此样例演示了在 TIM16 中基本计数功能,并使能了更新中断,每次重装 ARR 值时会产生一次更新中断,并在中断中翻转 LED 灯,LED 灯会以 5Hz 的频率进行翻转。

## 17 TIMER17

### 17.1 TIM17\_Counter

此样例演示了在 TIM17 中基本计数功能,并使能了更新中断,每次重装 ARR 值时会产生一次更新中断,并在中断中翻转 LED 灯,LED 灯会以 5Hz 的频率进行翻转。

## 18 TIMER3

### 18.1 TIM3\_GatedMode

此样例演示了 TIM3 从模式下的门控触发功能，配置 CH1(PA06)作为门控触发输入信号，并使能触发中断，每次进触发中断，翻转一次 LED 灯，在 IDE 仿真界面里，可以看到，当 PA06 输入低电平时，定时器 CNT 寄存器停止计数，当 PA06 输入高电平时，定时器 CNT 寄存器持续在计数。并且 PA06 引脚上的每产生一次触发中断，LED 都会翻转一次

### 18.2 TIM3\_InputCapture\_DMA

此样例演示了在 TIM3 输入捕获功能，捕获数据通过 DMA 传输到变量 CC1\_Capture 中，并通过串口打印输出。UART\_TX->PA2UART\_RX->PA3

### 18.3 TIM3\_SynchronizationGated

此样例演示了 TIM1 和 TIM3 同步触发（门控触发模式）功能，TIM1 作为主机，配置为比较输出功能，OC1 输出频率 1kHz，占空比 50%（即 1ms 高电平 1ms 低电平循环），TIM3 作为从机，配置为外部门控触发模式，TIM1 的 OC1 信号连接到 TIM3 作为 TIM3 的门控触发输入，配置 TIM3 的计数溢出周期为 125us，并允许溢出中断，且在溢出中断中对 LED 灯翻转。通过运行程序，通过逻辑分析仪可以看到，当 TIM1\_CH1(PA3)输出高电平时，LED(PB5)电平翻转；当 TIM1\_CH1(PA3)输出低电平时，LED(PB5)电平不翻转

### 18.4 TIM3\_SynchronizationReset

此样例演示了 TIM1 和 TIM3 同步触发（复位触发模式）功能，TIM1 作为主机，配置复位 TRGO 输出 (TIM1\_CR2.MMS=000)，TIM3 作为从机并且使能事件更新中断，配置收到主机的 TRGO 信号后复位 (TIM3\_SMCR.SMS=100)。循环打印 TIM3 计数值。

### 18.5 TIM3\_Update

此样例演示了在 TIM3 中基本计数功能，并使能了更新中断，每次重装 ARR 值时会产生一次更新中断，并在中断中翻转 LED 灯，LED 灯会以 5Hz 的频率进行翻转。

## 19 USART

### 19.1 USART\_HyperTerminal\_AutoBaud\_IT

此样例演示了 USART 的自动波特率检测功能，调试助手发送一个字符 0x7F，MCU 反馈字符串：Auto BaudRate Test。

### 19.2 USART\_HyperTerminal\_DMA

此样例演示了 USART 的 DMA 方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None,下载并运行程序后，通过 USART 会接收到 0x1-0xC,然后通过上位机下发 12 个数据，例如 0x1~0xC,则，MCU 会把接收到的数据再次发送。

### 19.3 USART\_HyperTerminal\_IT

此样例演示了 USART 的中断方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None,下载并运行程序后,上位机通过 USART 会接收到 0x1-0xC,然后通过上位机下发 12 个数据，例如 0x1~0xC,则，MCU 会把接收到的数据再次发送到上位机。

### 19.4 USART\_HyperTerminal\_Polling

此样例演示了 USART 的 POLLING 方式发送和接收数据，USART 配置为 115200，数据位 8，停止位 1，校验位 None,下载并运行程序后，通过 USART 会接收到 0x1-0xC,然后通过上位机下发 12 个数据，例如 0x1~0xC,则，MCU 会把接收到的数据再次发送。

## 20 WWDG

### 20.1 WWDG\_IT

此样例演示了 WWDG 的提前唤醒中断功能，看门狗计数器向下计数到 0x40 时产生中断，中断中喂狗，可以确保看门狗不会复位。

### 20.2 WWDG\_Window

此样例演示了 WWDG 的 窗口看门狗功能，配置 WWDG 的窗口上线（下限固定是 0x3F），程序中通过 delay 延时函数，确保程序是在 WWDG 计数窗口内进行喂狗动作，通过 LED 灯闪烁，可以判断窗口内喂狗并未产生复位。